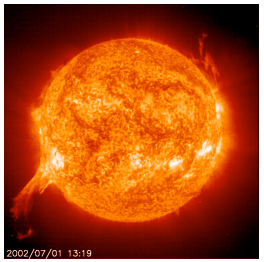
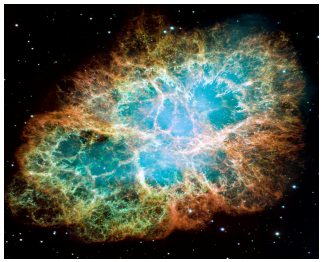
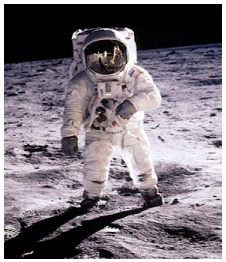
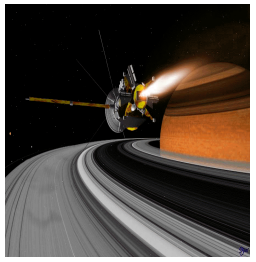
RAL Space

The Evolution of the GERB Ground Segment Processing System

Peter Allan, Head of Space Data Division

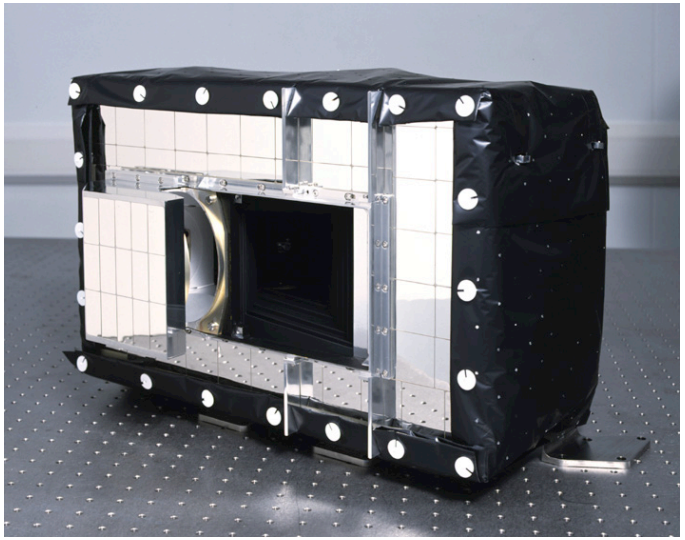





Overview

Need for long term operational system

Design choices

Changes along the way

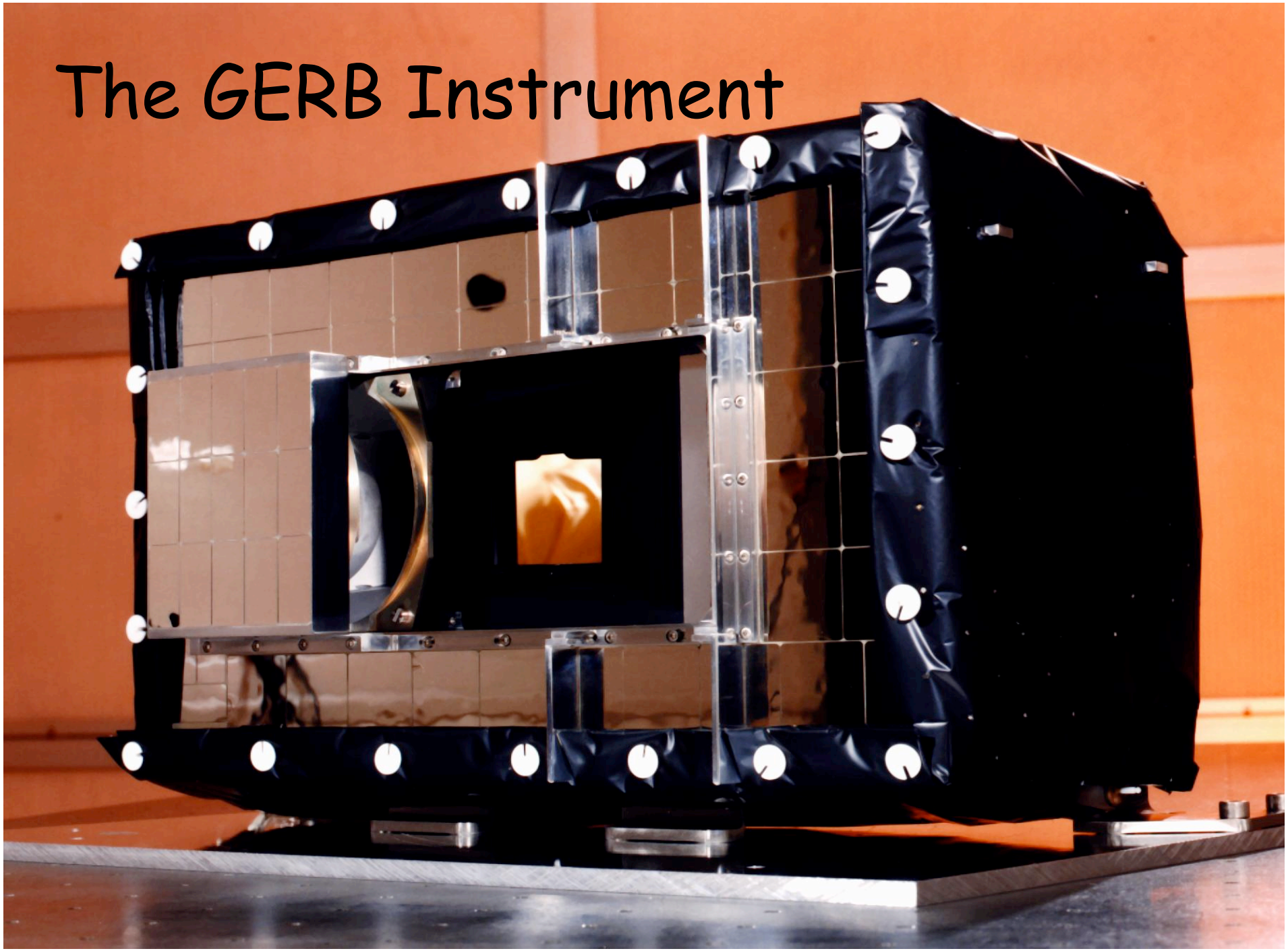


Misnomer

GGSPS is the system at RAL

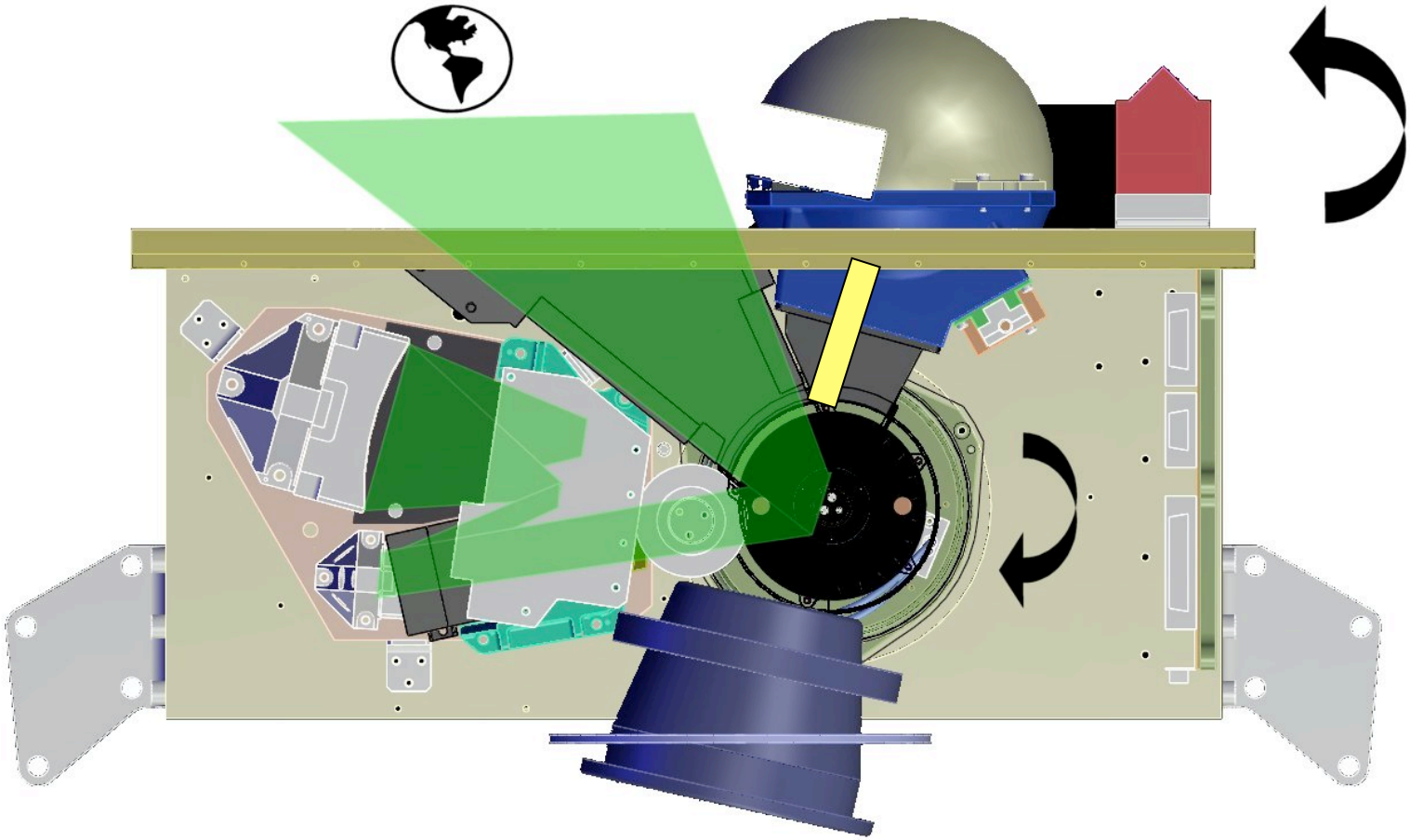
- Not the full ground segment

The GERB Instrument



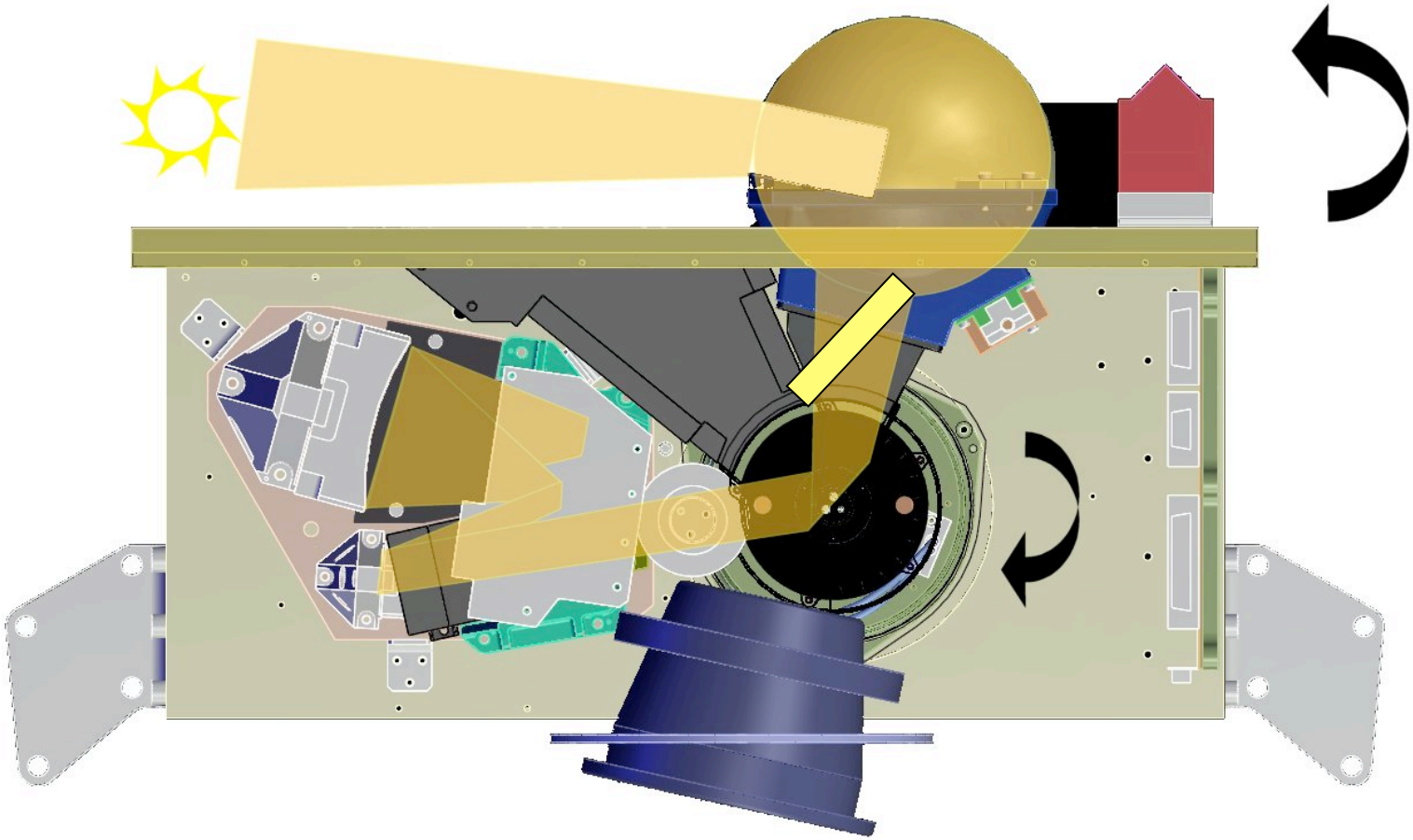
RAL Space

600 ms rotational cycle



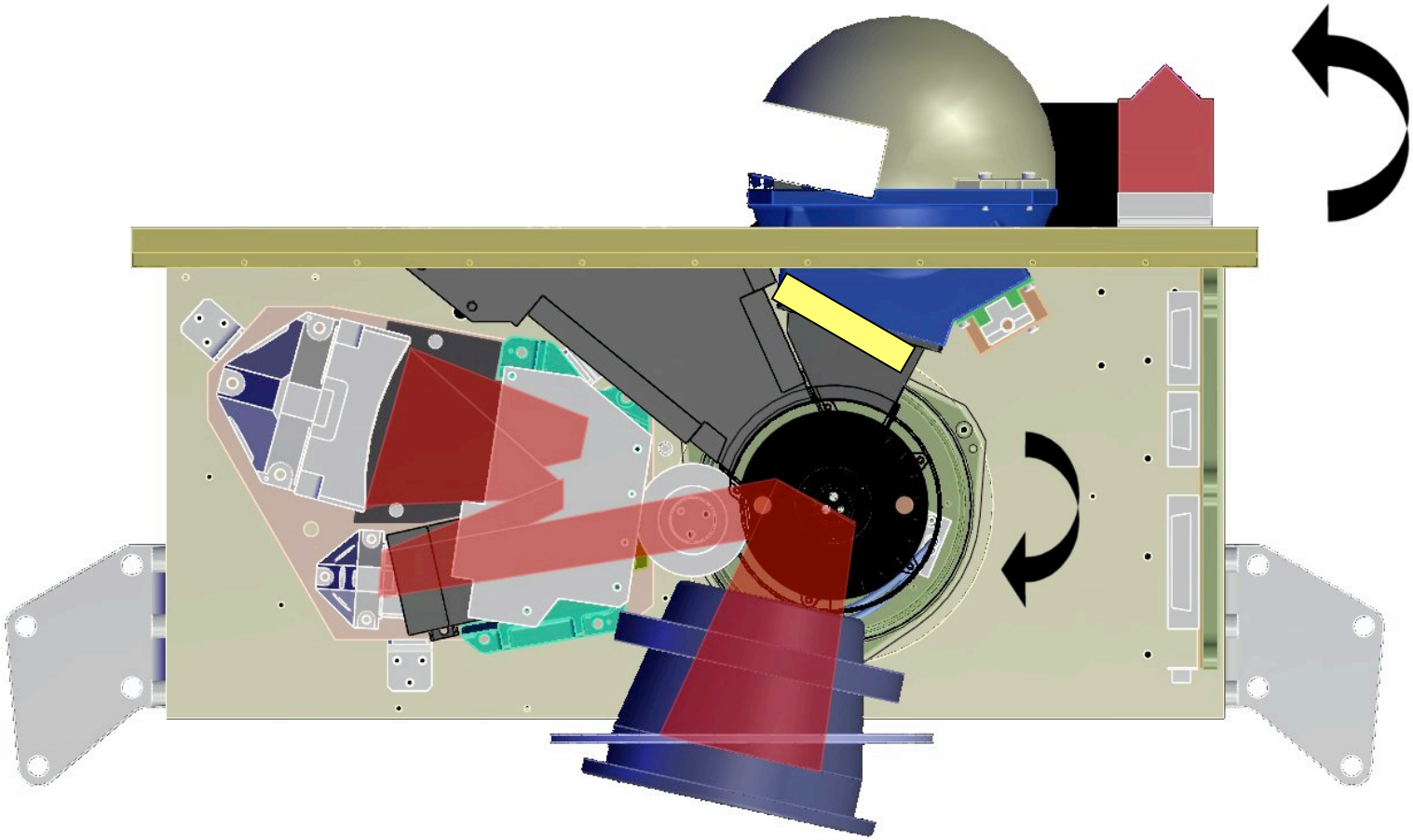
RAL Space

600 ms rotational cycle



RAL Space

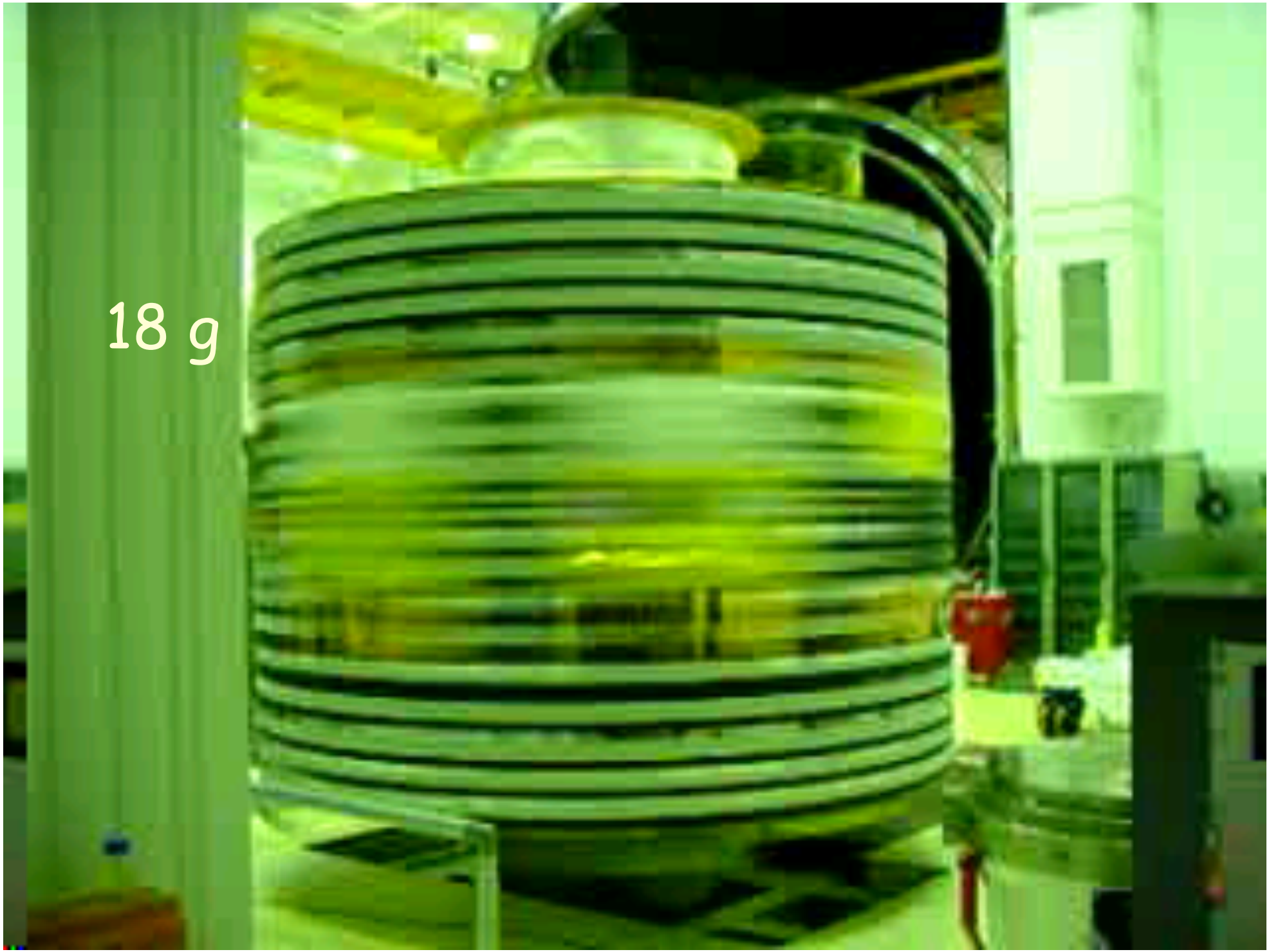
600 ms rotational cycle



S111E5238


3 to 4 g


Up to 9g


18 g

Players

EUMETSAT receive the raw data from MSG

RAL

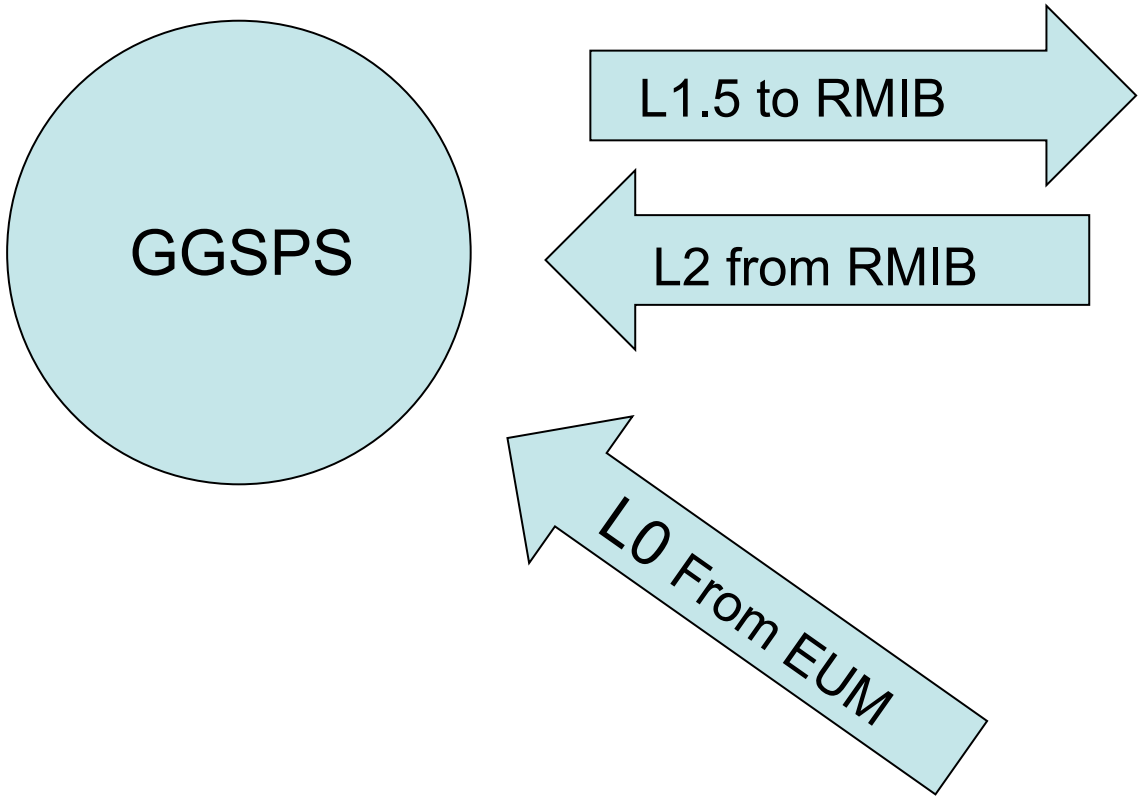
- Designed and led the building of the instrument
  - Calculate calibrated geolocated radiances

RMIB

- Expertise in deriving fluxes
  - Calculate fluxes

Imperial College – science and operations

Flow of data



Planning for the long term

- GGSPS was designed from 1996
- Originally planned launch in Oct 2000
- Actual launch of MSG-1 in 2002
- MSG-2 launched in December 2005
- MSG-3 launched recently (5 July 2012)
- Expected to go to at least 2018
- Well over 20 years from design to end
  - But at least we knew that!

Choosing the system

Candidate operating systems

- Various favours of (commercial) Unix
- Linux
- VMS
- Windows NT

Choice at end of 1996

- PSS

Considerations

Software development tools

Robust database (central to design of GGSPS)

Programming language

And the winner was …

- Digital Unix (-> Tru64)
- Ingres database
- C++

Slow evolution

Replace original Alpha hardware with more modern processors

Use NAS boxes for data storage

Changes to getting the raw data

Original scheme

- Leased line to EUM operational computers
- Received a data packet every 0.6 sec
- Driven by EUM need for security

Later, raw data delivered to U-MARF

- Now collect data files from U-MARF via internet
- Simpler
- Much cheaper

The Big Bang

HP

- which had merged with Compaq
  - which had bought Digital
    - announced it would drop support for Tru64

Options

- Buy enough hardware to last to the end of the mission
- Port to something with a future

Port to Linux

C++ is portable, isn't it?

Status of Ingres on Linux was uncertain

- Move database to Postgres

Porting process

- Quick build of code on Linux using Ingres as DB
- Careful port to give code that runs
  - Passes unit tests, executables don't crash
- Run full system test, compare data products

Problems along the way

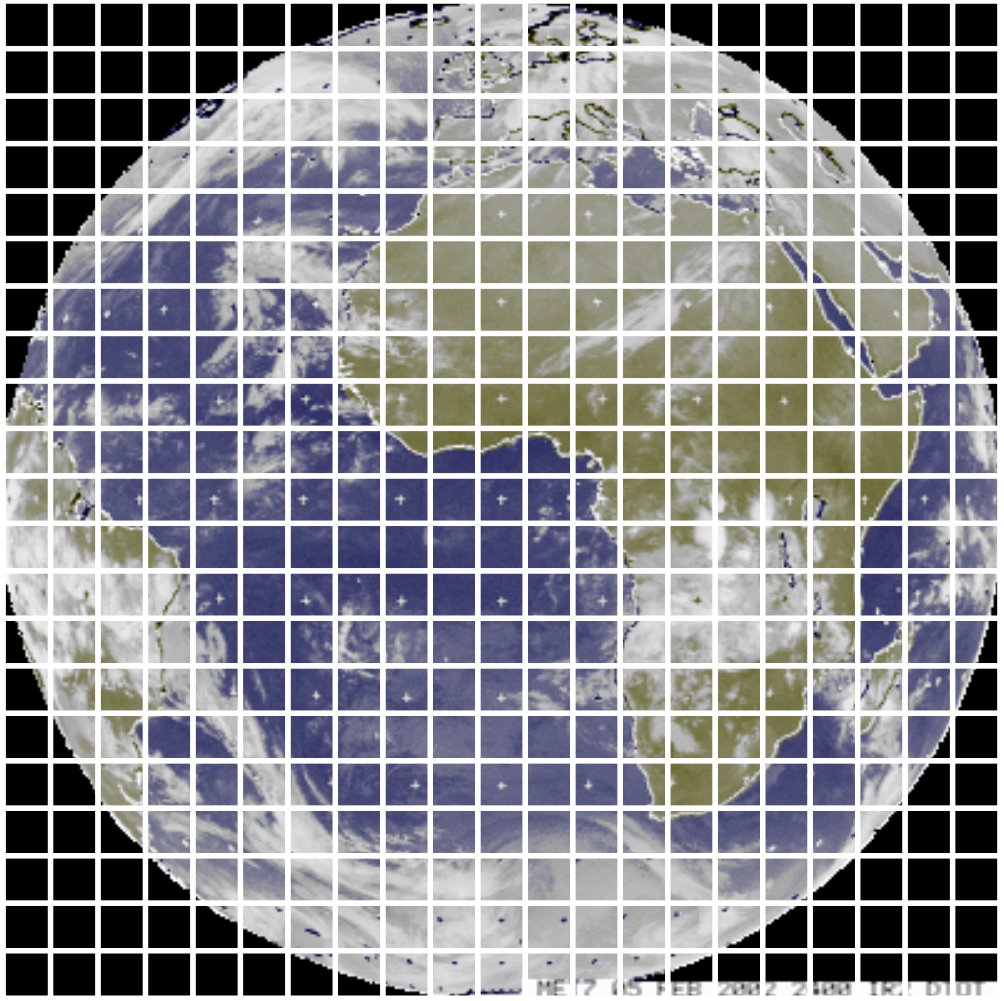
- Make files
- Compiler on Tru64 let us get away with some poor code
  - e.g. multiple fclose()
- Some data types in Ingres different in Postgres
- Error checking used in Ingres not available in Postgres

```
if ( fp  = fopen() ) {
   ...
   fclose(fp);
  }
// Final close “to be
   sure”
  fclose(fp);
}
```

Operational problems

Design of GERB is simple

- Scan E-W with no filter
- Scan W-E with short wave filter
- Repeat



But ...

- Position of detectors suffer from jitter
  - Timing signal from MSG
- Stray light affects images more than expected
- Mirror can stick on occasions
- Sensor swaps cause position offsets

-> Time spent checking data was much higher than expected

Operational Improvements

- Trap counting data
- Trap mirror pointing anomalies
- Automate generation of geo long-term trend plots
- Automate handling of stray light data
- Automate detection of anomalies
- Improve daily movie software
- Tools to compare L1.5 geo against RGP
- Improve robustness of generation of geo plots
- Extend range of quantities monitored in eng reports

Operational Improvements

Have reduced time spent on routine data validation

Can devote more time to improving the system

GERB-3

Mirror mechanism

- Velocity control on GERB-1 & GERB-2
- Position control on GERB-3 & GERB-4

New mechanism has led to revised data packet

Conclusions

- Philosophy of GERB has remained constant
- Have evolved the processing system to be more powerful and useful
- Have improved the operational tools to automate time consuming processes
- Set fair for GERB-3 and 4